

YASAIw.xla – A modified version of an open-source add-in for Excel to provide additional functions for Monte Carlo simulation.

By Greg Pelletier, Department of Ecology, P.O. Box 47710, Olympia, WA 98504-7710

(gpel461@ecy.wa.gov)

June 2009

Introduction

YASAI is a freely-available open-source add-in for Microsoft Excel that was developed at Rutgers University primarily to teach Monte Carlo simulation at the university level (Eckstein and Riedmueller, 2001). The basic functionality and theory of YASAI is presented in detail by Eckstein et al. (2000), Eckstein and Riedmueller (2002), and Eckstein and Riedmueller (2002).

We modified the original version 2.0 of YASAI (from the Rutgers Web site at <http://www.yasai.rutgers.edu/>) by adding several new functions and features as described in this document. The modified version of YASAI is called YASAIw version 2.0w. Some of the documentation provided by Eckstein and Riedmueller (2002) is repeated here to provide context for the new functions that we added in the modified version of YASAIw.

All of the original functions of YASAI are preserved in the new version. The following new features have been added in YASAIw:

- Additional distributions for specifying random variables.
- Functions for generating correlated random variables.
- Sensitivity analysis to estimate the correlation between forecasted variables and each assumed input variable and the contribution to variance of each forecast variable from each assumption variable.
- The ability to call user-defined macro VBA subroutines with each iteration of the simulation. This function allows the user to construct more complex models for prediction of the forecasted variables using VBA.
- The ability to use YASAIw functions in any or all of the worksheets in the workbook for the simulation.
- Additional output worksheets with cumulative distributions, the values of each variable during each iteration, and the results of sensitivity analysis.

Installing YASAIw

The current version of YASAIw is supported for Excel version 2007 or later. It should also work with Excel version 2003. Please follow these instructions to manually install YASAIw in Excel 2007 or later:

1. Download and copy to hard disk: The first step is to download the file YASAIw.xla from Ecology's FTP site or Web page, and copy it to the hard disk of the PC on which you wish to install YASAIw.
2. Check for and disable any old versions of YASAIw: Launch Excel. Select "Excel Options/Add-Ins/" from the Office button menu and then click on "Go" to manage add-ins. If "YASAIw" appears in the list of add-ins, uncheck it. Then click the "OK" button.
3. Configure Excel security: In Excel, select the "Developer" tab of the ribbon, then click on "Macros Security". Select "Enable all macros" security. Click OK.
4. Install the YASAIw add-in: Select "Excel Options/Add-Ins/" from the Office button menu and then click on "Go" to manage add-ins. Click "Browse...", locate the latest version of the file YASAIw.xla where you placed it in your hard disk, and click "Open". If Excel asks if it can copy the file, or if Excel asks if it can overwrite another file of the same name, click "Yes". If this fails then you should exit from Excel and manually copy the new YASAIw.xla file to the same location where Excel expects to find the old one. If YASAIw is installed, then "YASAI Simulation..." and "YASAI Charts..." should appear on Excel's "Tools" menu.

Existing functions in the original version of YASAI (version 2.0)

The following sub-sections of this section are copied or paraphrased mostly verbatim from Eckstein and Riedmueller (2002).

Design of YASAI

YASAI is designed in accordance with the following principles:

- The add-in is implemented as a single file in the Visual Basic for Applications (VBA) language. Any user on any computer equipped with a reasonably recent version of Excel should be able to use it, provided they are allowed to run Excel macros.
- The number of dialog boxes and steps in the user interface is kept to an absolute minimum. As many features as possible are controlled directly with functions that can be invoked from spreadsheet cell formulas, which minimizes the number of interface elements needed, and tends to afford users the maximum flexibility in combining the functions provided by the software. Running a simulation should produce full, printable output report in a single step.
- Beginning users should be able to easily construct and run simple models, without mathematical contortions such as inverting cumulative distributions.
- The code is open-source and freely available for any purpose (although, of course, there is no warranty). Users who want additional features should be able to add them and share them with the rest of the user community.

Specifying random variables

The original version 2.0 of YASAI distributed by Rutgers University provides several random variable generators that were found by Eckstein and Riedmueller, 2001 to be sufficient for their teaching needs:

GENUNIFORM(a, b): Returns a value uniformly distributed over the interval [a, b).

GENNORMAL(m, s): Returns a value with a normal distribution with mean m and standard deviation s.

GENBINOMIAL(n, p): Returns an integer drawn at random from a binomial distribution with n trials and probability p of success at each trial. If n = 0, then the return value is 0.

GENPOISSON(m): Returns a an integer chosen from a Poisson distribution with mean value m. If m is zero, so is the returned value.

GENTABLE(V, P): Allows a discrete distribution to be specified by an arbitrary table. The arguments V and P are blocks of cells or lists of values (for example, "{1,3,7}") having the same shape. The probability of choosing any particular value from V is equal to the corresponding value in P.

GENEXPON(a): Follows an exponential distribution with mean value 1/a.

GENGEOMETRIC(p): Returns a geometric random variables with a probability p of being 1. This variable is equal to the number of trials of a mean p Bernoulli (or equivalently, GENBINOMIAL(1,p)) variable until the value 1 is obtained. The value of p must be greater than 0, and less than or equal to 1, or an error value is returned.

GENTRIANGULAR(a, b, c): Returns a value from a triangular distribution with minimum a, mode b, and maximum c. The arguments must be numbers with the property $a < b < c$, or an error value is returned.

These functions may appear in cell formulas in arbitrarily nested and complicated ways. For example, the formula

```
=GENNORMAL(GENBINOMIAL(175,0.63),GENUNIFORM(1,4)) + GENPOISSON(5)
```

generates a normal random variable whose mean is in turn a random binomial variable of 175 trials with success probability 0.63, and whose standard deviation is in turn a uniform random variable between 1 and 4; to this value is added a Poisson variable with mean 5. Some commercial add-ins provide this level of flexibility, but others require the navigation of complicated dialog boxes to create random variables. We feel that the function-based design is simpler to teach, easier to use, and more flexible.

Specifying outputs

YASAI's method for specifying model outputs is very simple, and appears to be unique. To specify that output statistics should be tabulated for a model quantity, say x, one includes somewhere in the spreadsheet the formula SIMOUTPUT(x, name). Here, name is an argument that specifies what the output should be called. The SIMOUTPUT function always returns the value x. When working interactively with a spreadsheet, it does nothing more. However, when a simulation is running, SIMOUTPUT records each value it sees for later analysis.

For example, suppose that a model contains total revenue in cell B5, total cost in cell B6, and cell B7 computes profit via the formula =B5 – B6. If we wish the simulation to record total profit, we would replace the =B5 – B6 formula in B7 with

=SIMOUTPUT(B5 – B6,"Profit") .

SIMOUTPUT functions can be embedded into formulas in arbitrary ways. For example, in this same context, the formula

=SIMOUTPUT(SIMOUTPUT(B5,"Revenue") – SIMOUTPUT(B6,"Cost"),"Profit")

Simultaneously records the value of B5 – B6 as "Profit", the value of B5 as "Revenue", and the value of B6 as "Cost".

In a simpler example, suppose cell C6 contains a Poisson random variable whose mean value is specified by cell C3. While C6's main purpose is to serve as an input to other formulas in the model, we would like the simulation output to contain statistics about it. To ensure the output contains this information, we just change the formula in C6 from =GENPOISSON(C3) to, for example

=SIMOUTPUT(GENPOISSON(C3),"Poisson Input Variable") .

Creative use of conditional formulas can afford even greater flexibility. Suppose that cell D12 contains net cash flow. The formula

=SIMOUTPUT(ABS(D12),IF(D12>=0,"Profit","Loss"))

will create two outputs. The first, "Profit", will contain statistics on the net cash flow in D12, but tabulated only over those instances where it was nonnegative. The second, "Loss", will tabulate the absolute value of net cash flow, but only over those instances where net cash flow was negative.

In a further measure of flexibility, if multiple SIMOUTPUT expressions in a model have the same name argument, their values are pooled together in the simulation output.

We are not aware of another spreadsheet add-in that provides such broad flexibility in specifying outputs. Nevertheless, YASAI's design is simple to use and implement, and the added flexibility it affords is not a distraction to beginning students, who may just learn the simplest cases. Specifying an output does not require any special user interface elements. Furthermore, since the output specification is part of the ordinary formula mechanism, the spreadsheet cannot "forget" which cells are outputs. Similarly, examining the spreadsheet formulas reveals at a glance which cells are outputs — an advantage when trying to understand other people's models.

Specifying scenarios

In many applications of simulation, the actual objective is to evaluate a number of similar models, which vary only in some selected decision parameters. One may then wish to select the parameter combination that gives the highest expected profit or lowest expected cost, perhaps subject to some

acceptable level of risk quantified by other some other set of outputs. We call each tested set of parameter values a scenario.

Like some commercial add-ins, YASAI provides a mechanism for automating the comparison of scenarios. In keeping with the design philosophy of Eckstein and Riedmueller (2001), the mechanism is based on a function that can be embedded arbitrarily into cell formulas: `PARAMETER(L, k, name)`. The first argument, L, is a block of cells or a list specifying the possible return values. The name argument is a character string describing the parameter, and is used only in the output reports. The k argument specifies the number of scenarios between changes in the parameter. For example, a cell containing `=PARAMETER({1,2,3},1,"fred")` specifies a parameter called "fred" that will change every scenario, first taking the value 1, then 2, and then 3. A cell containing the formula `=PARAMETER({1,8,2.3,2},2,"daisy")` defines a parameter called "daisy" that takes the value 1 for the first two scenarios, then 8 for the next two, 2.3, for the next two, and -2 for the last two. The argument k makes it possible to try combinations of values of various parameters without having to write every combination explicitly in a list (although that approach is possible). For instance, suppose that the cells E6:E8 contain the formulas

```
=PARAMETER({0,1},1,"A")  
=PARAMETER({100,150,200},2,"B")  
=PARAMETER({20,40},6,"C") ,
```

signifying parameters called A, B, and C, respectively. Then, if we run 12 scenarios, the parameters would assume the following values:

Scenario	A	B	C
1	0	100	20
2	1	100	20
3	0	150	20
4	1	150	20
5	0	200	20
6	1	200	20
7	0	100	40
8	1	100	40
9	0	150	40
10	1	150	40
11	0	200	40
12	1	200	40

Thus, all 12 possible parameter combinations would be tested, without having to code an exhaustive list of combinations into the spreadsheet.

When a model is being used interactively, the values of PARAMETER values change every k recalculations, so that the user can easily observe the model recalculating with all the conceivable parameter combinations.

New functions in YASAIw.xla version 2.0w

This section describes new functions that have been added to the original YASAI in our modified version 2.0w of YASAIw.

Specifying random variables in YASAIw

In addition to the functions in the original version 2.0 of YASAI, five new functions are provided in YASAIw for specifying random numbers from various distributions as follows.

=GENLOGNORMAL(m, s): Returns a value with a lognormal distribution with mean m and standard deviation s.

=GENCFD(range): returns a value that is randomly selected from a cumulative frequency distribution (CFD) that is specified in a contiguous two column range of Worksheet cells, where the first column of the range contains the frequency values between 0 and 1 (which must begin with 0 and end with 1 and be sorted from 0 to 1), and the second column of the range specifies the cumulative distribution values for each frequency. For example, if cells AZ26:BA36 are cells that contain the data for a CFD as follows (e.g. min=1490, 5%tile=1540, etc.), then AZ26:BA36 would be entered as the range argument:

0	1490
0.05	1540
0.1	1590
0.15	1622.5
0.25	1682.5
0.5	1820
0.75	1962.5
0.85	2042.5
0.9	2105
0.95	2190
1	2275

=GENLIMITNORMAL(m, s, min, max, optional method): Returns a value with an underlying normal distribution with mean m and standard deviation s that is bounded by the min and max. The 'method' argument is optional and allows for calculation using one of the following two methods:

- Method=0 (or missing) samples according to the underlying distribution until a sampled value is within the bounds of the min and max. This method changes the percentiles relative to the underlying distribution and takes the density of the underlying distribution conditioned on being within the truncation bounds.
- Method 1 samples from an unbounded distribution and replaces any values that are above or below the max or min with the max or min. This method preserves the same percentiles as the unbounded distribution within the bounds of the min and max.

=GENLIMITLOGNORMAL(m, s, min, max, optional method): Returns a value with an underlying lognormal distribution with mean m and standard deviation s that is bounded by the min and max. The 'method' argument is optional and is the same as described for GENLIMITNORMAL.

=GENBETAPERT(min, mode, max, optional weight): Returns a value with a PERT-beta distribution with a minimum (min), most likely value (mode), and maximum (max). The optional parameter (weight) is a weighting factor that is used to estimate the mean value (weight=4 is assumed if weight is not specified). As the weight parameter is increased the estimated mean value approaches the most likely value (mode). The PERT-beta is useful for modeling expert opinion in which a variable is bounded with known or estimated bounds (min and max) and for which a "most likely value" is known or can be estimated. The PERT-beta distribution is a smoother curved alternative to a triangular distribution. The PERT-beta distribution $\beta_p(\text{min}, \text{mode}, \text{max}, \text{weight})$ is related to the standard beta distribution $\beta(\alpha_1, \alpha_2)$ through the following relationships (EPA, 2007; Vose, 2009):

$$\begin{aligned}\beta_p(\text{min}, \text{mode}, \text{max}, \text{weight}) &= \text{min} + (\text{max} - \text{min}) * \beta(\alpha_1, \alpha_2) \\ \alpha_1 &= ((\mu - \text{min}) * (2 * \text{mode} - \text{min} - \text{max})) / ((\text{mode} - \mu) * (\text{max} - \text{min})) \\ \alpha_2 &= (\alpha_1 * (\text{max} - \mu)) / (\mu - \text{min}) \\ \mu(\text{mean}) &= (\text{min} + \text{weight} * \text{mode} + \text{max}) / (\text{weight} + 2)\end{aligned}$$

The mean value (μ) of the PERT-beta distribution is assumed using the equation above in order to estimate the appropriate shape parameters (α_1 and α_2) of the corresponding beta distribution. Unlike the triangular distribution, the mean value (μ) of the PERT-beta distribution is more influenced by the most likely value (mode) depending on the value of the weight parameter. The weight parameter influences the spread of the distribution around the most likely value, with larger values of weight resulting in narrower spread. A default weight = 4 is suggested based on standard PERT network assumption that the best estimate for the duration of a task = $(\text{min} + 4 * \text{mode} + \text{max}) / 6$ (Vose, 2009).

Correlated variables

Generating two sequences of random numbers from standardized normal distributions with a given linear correlation is done in two steps in YASAIw:

1. Generate two sequences of uncorrelated standardized normal variables (e.g. X1 and X2)
2. Define a new sequence of correlated standardized normalized variable $Y1 = r * X1 + (1 - r^2)^{0.5} * X2$, where r is the correlation coefficient between Y1 and X1.

Generating two sequences of random numbers from uniform distributions with a given linear correlation is done in three steps in YASAIw:

1. Generate two sequences of uncorrelated standardized normal variables (e.g. X1 and X2)
2. Define a new sequence of correlated standardized normalized variable $Y1 = \text{radj} * X1 + (1 - \text{radj}^2)^{0.5} * X2$, where 'radj' is the calculated linear correlation coefficient between Y1 and X1 estimated by adjusting the assumed linear correlation 'r' between uniform variables that are used to sample from the non-parametric distribution. This new sequence for Y1 is a standard

normal variable with a linear correlation 'radj' with the X1 sequence, where $\text{radj} = 2 * \sin(\pi/6 * r)$. Note that r is also the Spearman rank correlation between the standardized normals as well as being the linear correlation between the uniform variables.

3. Calculate the new sequence of linearly correlated uniform variables (0 to 1) using Excel's NORMSDIST function that returns the probability (0 to 1) that a standardized normal variable will be less than or equal to Y1.

Correlated standard normal variables are used by YASAIw to generate correlated normal or lognormal variables. Correlated uniform variables are used to generate correlated nonparametric variables with specified CFDs.

Nine functions are provided for use in generating correlated random variables from either normal or lognormal distributions.

=GENZ(): Generate independent standardized normal variables

=GENNORMALX(z1, m, s): Generate independent normal variable with m=mean, s=standard deviation, using z1= independent standardized normal variable.

=GENNORMALY(z1, z2, r, m, s): Generate normal variable with m=mean, s=standard deviation, correlated to the independent variable by r= correlation, using z1 and z2= independent standardized normal variables.

=GENLOGNORMALX(z1, m, s): Generate independent log-normal variable with m=mean, s=standard deviation, using z1= independent standardized normal variable.

=GENLOGNORMALY(z1, z2, r, m, s): Generate lognormal variable with m=mean, s=standard deviation, correlated to the independent variable by r= correlation between variables, using z1 and z2= independent standardized normal variables.

=GENLIMITNORMALX(z1, m, s, min, max): Generate truncated independent normal variable with m=mean, s=standard deviation, using z1= independent standardized normal variable, min and max= minimum and maximum values of the truncated distribution. The truncation method is method 1 a described above.

=GENLIMITNORMALY(z1, z2, r, m, s, min, max): Generate truncated normal variable with m=mean, s=standard deviation, correlated to the independent variable by r= correlation, using z1 and z2= independent standardized normal variables, min and max= minimum and maximum values of the truncated distribution. The truncation method is method 1 a described above.

=GENLIMITLOGNORMALX(z1, m, s, min, max): Generate independent log-normal variable with m=mean, s=standard deviation, using z1= independent standardized normal variable, min and max= minimum and maximum values of the truncated distribution. The truncation method is method 1 a described above.

=GENLIMITLOGNORMALY(z1, z2, r, m, s, min, max) : Generate lognormal variable with m=mean, s=standard deviation, correlated to the independent variable by r= correlation between variables, using z1 and z2= independent standardized normal variables, min and max= minimum and maximum values of the truncated distribution. The truncation method is method 1 described above.

Two additional functions are provided for use in generating correlated random variables from nonparametric cumulative distribution functions (CFDs):

=GENCFDX(range, z1): returns an independent X value that is randomly selected from a cumulative frequency distribution (CFD) that is specified in a contiguous two column range of Worksheet cells (as described above for GENCFD). The second argument refers to a worksheet cell containing a GENZ function that is a standardized normal variable that is used to derive the uniform distribution value that is used to sample from the CFD.

=GENCFDY(range, z1, z2, r): returns a Y value that is rank-correlated to the X value and is randomly selected from a cumulative frequency distribution (CFD) that is specified in a contiguous two column range of Worksheet cells (as described above for GENCFD). The second and third arguments (z1 and z2) refer to worksheet cells containing independent GENZ functions that are used to estimate a correlated standardized normal variable that is used to derive a linearly correlated uniform distribution value that is used to sample from the CFD. The fourth argument is the linear correlation between the uniform variables that are used to derive the independent and dependent X and Y values from the CFDs (r is also the Spearman rank correlation between the standardized normals and the nonparametric random variables).

The following are instructions for generating normal random values of a dependent variable that are correlated to an independent variable for use as assumptions in Monte Carlo simulation with YASAIw (Figure 1):

1. Use =GENZ() to generate uncorrelated independent standardized normal variables in worksheet cells for the independent and the dependent variables (e.g. cell C4 and C5 in an example worksheet shown in Figure 1)
2. Enter the mean "m" and standard deviation "s" for the independent variable in worksheet cells (e.g. cells E4 and F4 in this example worksheet)
3. Enter the correlation coefficient "r", mean "m", and standard deviation "s" of the dependent correlated variable in worksheet cells (e.g. cells D5, E5, and F5 in this example worksheet), where r^2 represents the fraction of the variance of the dependent variable that is explained by variance of the independent variable, m is the mean of the correlated dependent variable, and s is the standard deviation of the correlated dependent variable.
4. Use =GENNORMALX(z1, m, s) to generate normal random values for the independent variable, where z1 references the worksheet cell where the =GENZ() function was used to generate the standardized normal variable for the independent variable (e.g. cell G4 in this example worksheet). For lognormal variables use =GENLOGNORMALX(z1, m, s) (e.g. cell K4).

- Use =GENNORMALY(z1, z2, r, m, s) to generate normal random values for the independent variable, where z1 references the worksheet cell where the =GENZ() function was used to generate the standardized normal variable for the independent variable, z2 is the =GENZ() for the dependent variable, r is the correlation coefficient of the independent to the dependent variable, m is the mean value of the dependent variable, and s is the standard deviation of the dependent variable (e.g. cell G5 in this example worksheet). For lognormal variables, use =GENLOGNORMALY(z1, z2, r, m, s) (e.g. cell K5).

The independent and dependent variables may also be from different parametric or nonparametric distributions. For example, the independent variable may be a nonparametric distribution (GENCFDX) and the dependent variable may be a normal distribution (GENNORMALY) or vice versa (GENNORMALX and GENCFDY). In these cases of mixed distribution types the correlation between the independent and dependent variables refers to Spearman's rank correlation.

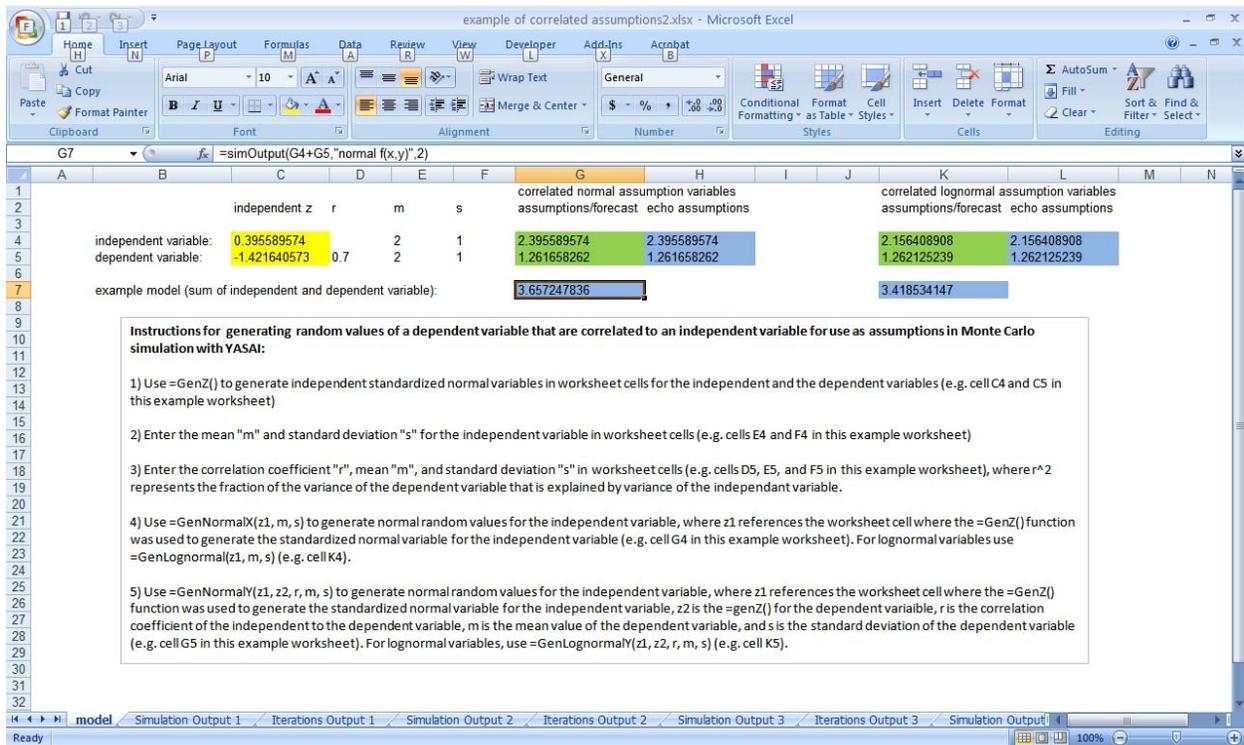


Figure 1. Example Worksheet using correlated variables.

Sensitivity analysis

Sensitivity analysis provides output of a Worksheet with Spearman's rank correlation coefficient for each pair of forecasts/assumptions for the SIMOUTPUT variables. In practice it is most useful to define SIMOUTPUT variables as either "echoes of the assumptions" or "forecasts" using a new optional third argument for the SIMOUTPUT function (1=echo of assumption, 2=forecast). "Assumptions" can be thought of as the cells that contain functions for generating randomized variables (e.g. using the GENNORMAL or other functions to specify random variables) that are used as inputs to the model. The "echoes of the assumptions" are cells that contain SIMOUTPUT functions that refer to the assumptions

as their argument so that the assumption values will be saved for the sensitivity analysis or to view their cumulative frequency distributions. The “forecast” cells are the model calculations that depend on the assumptions, and they contain SIMOUTPUT functions that use the model equations as their argument.

Using the example shown in Figure 1, the input variables X and Y are the model assumptions that are contained in cells G4 and G5, respectively, and they are echoed as SIMOUTPUT assumptions by entering the following functions in cells H4 and H5, respectively (note the optional third argument is 1 to specify these SIMOUTPUT functions as echoes of assumptions):

```
=SIMOUTPUT(G4, “X”, 1)
```

```
=SIMOUTPUT(G5, “Y”, 1)
```

There typically are several variables that are assumptions in any model, and any of these can also be echoed in the same way. The model result that depends on these assumptions can be thought of as a “forecast” that depends on the “assumptions”. For example, if the model calculation is in cell G7 and it is the sum of two numbers, X in cell G4 and Y in cell G5, then the model result is specified as a SIMOUTPUT forecast in cell G7 as follows (note the optional third argument is 2 to specify this SIMOUTPUT as a forecast):

```
=SIMOUTPUT(G4+G5, “X+Y”, 2).
```

If the optional third argument is not used (or is a value other than 1 or 2) then the SIMOUTPUT function will not be included in the sensitivity analysis.

The sensitivity analysis presents the Spearman’s rank correlation coefficient between each forecast with each assumption. The sensitivity analysis also presents the “contribution to variance” that each assumption provides for each forecast. Contribution to variance is estimated by summing the squared rank correlation coefficients for each forecast, and then taking the ratio of the squared rank correlation coefficient of each assumption to the sum of the squared rank correlation coefficients for each forecast (Vose, 2009).

Calling user-defined macro VBA subroutines during each iteration

YASAIw allows the option to call user-defined macro VBA subroutines with each iteration of the simulation. This allows the user to construct more complex models for prediction of the forecasted variables using VBA. For example, assumption variables that are randomly generated may be used as inputs to models that are executed with user-defined VBA subroutines or macros. The user has the option of running macros either before or after the workbook is recalculated, or both.

The YASAIw functions for specifying random variables generate a new set of random values each time the workbook is recalculated. During each iteration YASAIw forces a full calculation of the data in all of the worksheets of the active workbook. In some cases the user may want to run a macro before the workbook is recalculated in order to provide outputs that will be used to generate the new set of variables. In other cases the user may want to run a macro after the workbook is recalculated in order to

use the new set of assumption variables for input to a complex model that generates output that will be used as forecasts or predictions.

To run macros during simulation the user must include Subroutines in a VBA module of the active workbook with one or both of the following names:

- Public Sub YASAIwBeforeRecalc() – if a subroutine with this name is present in a VBA module of the active workbook then it will be run during each iteration before the workbook is recalculated by YASAIw.
- Public Sub YASAIwAfterRecalc() – if a subroutine with this name is present in a VBA module of the active workbook then it will be run during each iteration after the workbook is recalculated by YASAIw.

As an example of the use of user-defined macros, suppose that the user has a Subroutine named “RunQ2K” that needs to be run during each iteration after the workbook is recalculated by YASAIw to use the new set of assumptions as inputs to a complex model. The following subroutine would need to be present in a VBA module of the active workbook:

```
Public Sub YASAIwAfterRecalc()  
    Call RunQ2K  
End Sub
```

The following VBA statements should generally not be used in any user-defined VBA subroutines or functions that are executed during the simulation because they cause the YASAIw functions that are present in the worksheets to generate a new set of random variables, which will result in incorrect results of sensitivity analysis:

```
ActiveSheet.Calculate  
Application.Calculate  
Application.CalculateFull  
Application.CalculateFullRebuild
```

Saving results of user-defined macro VBA subroutines during each iteration

To specify that output statistics should be tabulated for an output of a user-defined VBA subroutine or macro in a worksheet cell, one includes somewhere in a worksheet the formula as follows:

=VBAOUTPUT(x, name, optional sensitivity)

Here, x is the address of the cell where the output of the VBA subroutine is located, name is an argument that specifies what the output should be called, and the optional sensitivity argument, if used, is set to either 1 (use the output as an assumption for sensitivity analysis) or 2 (use the output as a forecast for sensitivity analysis). The VBAOUTPUT function always returns the value in cell x. When working interactively with a spreadsheet, it does nothing more. However, when a simulation is running, VBAOUTPUT records each value it sees for later analysis.

For example, suppose that a user-defined VBA subroutine runs a water quality model of a river that writes an output value of the predicted dissolved oxygen concentration at river mile 3.5 into cell B5, and we want to name that variable “DO at RM 3.5” and use it as a forecast variable for the sensitivity analysis. To do this the following formula would be entered in a worksheet cell:

```
=VBAOUTPUT(B5,"DO at RM 3.5", 2)
```

It is possible to save outputs from user-defined subroutines using the SIMOUTPUT function, but this should be avoided because it would cause incorrect results of the sensitivity analysis. The user should also avoid placing any YASAIw functions in cells that the user-defined subroutines will use as output because the contents of these cells would be replaced with output values during each iteration.

Use of multiple worksheets in the workbook for the simulation model

The original version of YASAI required that all of the functions for the simulation model needed to be on a single worksheet, and that the simulation had to be run from that worksheet. In YASAIw we have changed this to allow the user to place YASAIw functions for the simulation model in any cells of any worksheets in the workbook. The simulation can be started from any worksheet with YASAIw. This allows the user to use any or all of the worksheets in the workbook for the simulation model. We also recommend that a workbook should be used for only one simulation model with YASAIw.

Running the simulation

To run a simulation, one selects “YASAI Simulation” from Excel’s Add-ins menu, which brings up YASAIw’s main dialog box, shown in Figure 2.

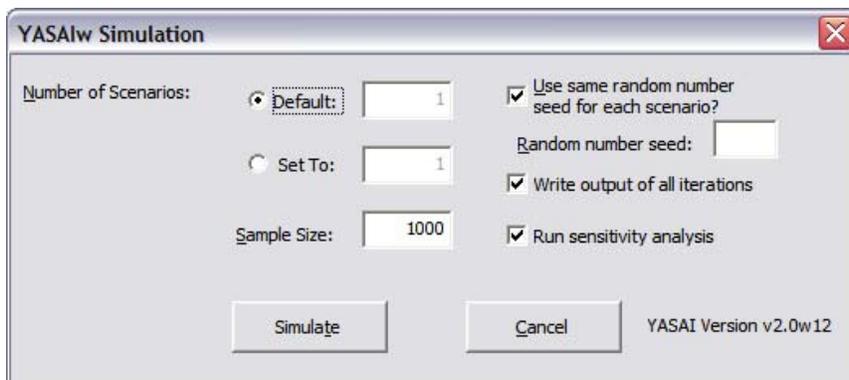


Figure 2. YASAIw’s main dialog box.

The dialog box allows the user to specify the number of scenarios and the number of sample recalculations per scenario. Ordinarily, nothing else needs to be entered. There is an option, however, to specify a fixed random number seed value (important when trying to exactly reproduce simulation behavior, and useful for model debugging). If the random seed field is not filled, YASAIw constructs a seed from the system clock. A final option, turned on by default in accordance with standard practice, causes the same random seed value to be used for all scenarios.

Two other check boxes are also present in the main dialog box and by default they are activated:

- “Write output of all iterations” provides output of a Worksheet called “Iterations Output” that gives output of the values of each value for all SIMOUTPUT functions.
- “Run sensitivity analysis” provides output of a Worksheet called “Sensitivity Output” with Spearman’s rank correlation coefficient for each pair of forecasts/assumptions for the SIMOUTPUT or VBAOUTPUT variables as described above. The sensitivity analysis can take a long time (e.g. many hours) for models with large numbers of pairs of forecasts and assumptions and large numbers of iterations. Therefore we provide the option to turn off the sensitivity analysis with this checkbox in the main dialog box to allow for faster run times if the user is not interested the sensitivity analysis.

Clicking the “Simulate” button on the dialog box starts the simulation. YASAIw recalculates the spreadsheet NS times, where N is the sample size and S is the number of scenarios, storing the x arguments of all SIMOUTPUT and VBAOUTPUT functions, organized by output name and scenario number. During this process, YASAIw shows a moving progress bar, and the user may abort the simulation by clicking an “Abort” button.

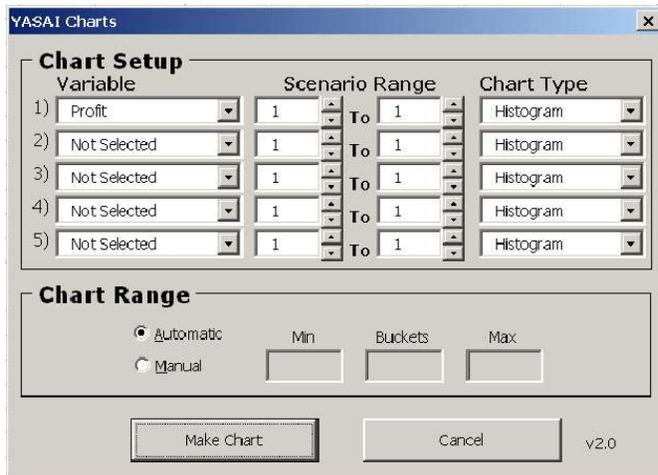
When all NS recalculations are complete, YASAIw produces at least two and up to four worksheets with output reports. The output reports become new worksheets of the current workbook, as follows:

- Simulation Output – this worksheet contains information for each output name/scenario number pair. For each such pair, the report contains the mean, standard deviation, minimum, maximum, and selected percentiles in 1% intervals.
- CFD Output – This worksheet contains output of the cumulative frequency distributions (CFD) for each output name/scenario number pair. For each such pair, the report contains the mean, standard deviation, minimum, maximum, and the CFD at 1% intervals
- Iterations Output – This optional output worksheet contains the values of all of the SIMOUTPUT variables for each iteration. This worksheet is written if this option is selected in the YASAIw main dialog box.
- Sensitivity Output – this optional worksheet contains the results of the Spearman’s rank correlation coefficient and the contribution to variance from each SIMOUTPUT or VBAOUTPUT assumption for each SIMOUTPUT or VBAOUTPUT forecast. This worksheet is written if this option is selected in the YASAIw main dialog box.

The reports can be immediately printed via Excel’s standard printing mechanism. To delete them, one just deletes the corresponding spreadsheet ply. The ‘YASAI Charts’ option in the Excel Add-In toolbar is not fully supported in this version.

Charting output from YASAIw

YASAIw can produce charts of simulation outputs. You must run simulation before trying to produce charts. To make a chart, select "YASAI Charts..." from the "Add-Ins..." menu, which produces a charting window:



You may chart up to five blocks of outputs, each corresponding to one row in the window. The first column selects which variables to chart. The second and third columns allow you to specify a range of scenarios for the selected variable, for example scenarios 1 through 5; a graph for each scenario in the range will appear in the output. The last column selects the kind of graph desired for the block of variables: "Histogram" is a standard bar chart, and "Cumulative plot" produces empirical cumulative distributions. You may mix the two kinds of graphs on a single chart. With the "automatic" button set under "Chart Range" YASAIw chooses the horizontal axis range and subdivisions to attempt to produce an attractive chart. "Manual" lets you specify a range from "Min" to "Max", with "Buckets" subdivisions.

YASAIw charts are regular Excel charts. Once they have been created, you modify them to suit your needs. The chart and its associate data each become a new worksheet ply. You may delete them if they are no longer needed. YASAIw charts are also "static" -- they are based on the simulation immediately preceding their creation; if you run another simulation of the same model, they will not automatically update. You must run a simulation again and make new charts if you want your charts to reflect a change to your simulation model.

Due to internal limitations in Excel, cumulative graphs are not possible for outputs with more than 32,760 observations. Histograms are possible for any sample size.

Known Issues

Updating Links

When you open a workbook containing YASAIw functions, Excel may display the message that it contains "automatic links to another workbook", and asks if you want to update the links. As of YASAIw version 2.0w17, it should not matter what you answer here -- YASAIw should automatically repair the links. Link updating generally occurs when you move a YASAIw-based spreadsheet between two computers where YASAIw is installed in different places in the file system.

If you get the following message when opening a workbook that contains YASAIw functions, click on "Continue" to leave the links as is and open the file:



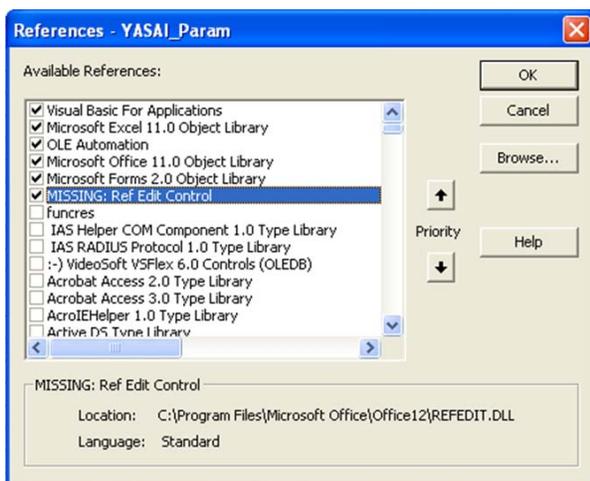
On some Excel installations, however, Excel does not seem to generate an "open workbook" event, and link updating is skipped. If link updating fails, your spreadsheet formulas may contain weird strings like "'!C:Documents and Settings\JoeUser\...\YASAIw.xla!:". If you delete these strings manually or with Excel's "Replace" function, things should start working normally again.

"YASAI Simulation..." appears on the Tools menu, but the some or all of the YASAIw spreadsheet functions evaluate to "#NAME!"

Excel's security settings may be preventing YASAIw from running. Select "YASAI Simulation..." from the Tools menu. You should immediately see a dialog box asking if you wish to enable macros. Click "Yes" or "Enable Macros". The YASAIw dialog box should appear. Click "Cancel". For each cell containing a YASAIw spreadsheet function, click the cell, place the cursor at the end of the formula in the formula bar, click, and then hit the enter key.

"Microsoft Visual Basic - Can't find project or library" and "MISSING: Ref Edit Control"

If this error message appears, open the VBA editor in Excel with Alt-F11 (it may already be open), and go to Tools/References for YASAI_Param and uncheck the reference to Ref Edit Control (and any other references that say "missing"). Then File/Save YASAIw.xla from the VBA editor. You may need to exit Excel and then restart Excel after saving YASAIw.xla for the change to become effective.



Excel 2007

Some users have reported cases when charts appear mangled in Excel 2007. Also, workbooks saved in backward-compatible formats from Excel 2007 may cause YASAIw crashes when opened in older versions of Excel.

References

Eckstein, J., S.T. Riedmueller, and S. Reed. 2000. YASAI: Yet Another Simulation Add-In for Excel. Version 2.0 User Guide. <http://www.yasai.rutgers.edu/yasai-guide-20.html>

Eckstein, J. and S.T. Riedmueller. 2001. YASAI: Yet another addin for teaching elementary Monte Carlo simulation in Excel. Rutcor Research Report RRR 27-2001, APRIL, 2001, Rutgers University, Piscataway, New Jersey.

Eckstein, J. and S.T. Riedmueller. 2002. YASAI: Yet another addin for teaching elementary Monte Carlo simulation in Excel. *Informations Transactions on Education*, Volume 2, Number 2, January 2002.

<http://archive.itejournal.informs.org/Vol2No2/EcksteinRiedmueller/>

EPA. 2007. EPA: Pesticides - A Discussion with the FIFRA Scientific Advisory Panel Regarding the Terrestrial and Aquatic Level II Refined Risk Assessment. Models (Version 2.0) - Chapter III. Terrestrial Level II Model. http://www.epa.gov/oppefed1/ecorisk/fifrasap/rra_chap_three.htm

Vose. 2009. ModelAssist for Crystal Ball. Vose Consulting, Gent, Belgium. <http://www.vosesoftware.com>